A User-centric Evaluation of Recommender Algorithms for an Event Recommendation System

Simon Dooms
Wica-INTEC, IBBT-Ghent University
G. Crommenlaan 8 box 201
B-9050 Ghent, Belgium
Simon.Dooms@UGent.be

Toon De Pessemier
Wica-INTEC, IBBT-Ghent University
G. Crommenlaan 8 box 201
B-9050 Ghent, Belgium
Toon.DePessemier@UGent.be

Luc Martens
Wica-INTEC, IBBT-Ghent University
G. Crommenlaan 8 box 201
B-9050 Ghent, Belgium
Luc1.Martens@UGent.be

ABSTRACT

While several approaches to event recommendation already exist, a comparison study including different algorithms remains absent. We have set up an online user-centric based evaluation experiment to find a recommendation algorithm that improves user satisfaction for a popular Belgian cultural events website. Both implicit and explicit feedback in the form of user interactions with the website were logged over a period of 41 days, serving as the input for 5 popular recommendation approaches. By means of a questionnaire users were asked to rate different qualitative aspects of the recommender system including accuracy, novelty, diversity, satisfaction, and trust.

Results show that a hybrid of a user-based collaborative filtering and content-based approach outperforms the other algorithms on almost every qualitative metric. Correlation values between the answers in the questionnaire seem to indicate that both accuracy and transparency are correlated the most with general user satisfaction of the recommender system.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; H5.2 [**User Interfaces**]: User-centered design

General Terms

Algorithms, Experimentation, Human Factors.

Keywords

Recommender systems, events, user-centric evaluation, experiment, correlation, recommendation algorithms.

1. INTRODUCTION

More and more recommender systems are being integrated with web based platforms that suffer from information overload. By personalizing content based on user preferences, recommender systems assist in selecting relevant items on these websites. In this paper, we focus on evaluating recommendations for a Belgian cultural events website. This website contains the details of more than 30,000 near future and ongoing cultural activities including movie releases, theater shows, exhibitions, fairs and many others.

In the research domain of recommender systems, numerous studies have focused on recommending movies. They have been studied thoroughly and many best practices are known. The area of event recommendations on the other hand is relatively new. Events are so called *one-and-only* items [5], which makes them harder to recommend. While other types of items generally remain available (and thus recommendable) for longer periods of time, this is not the case for events. They take place at a specific moment in time and place to become irrelevant very quickly afterwards.

Some approaches towards event recommendation do exist. For the Pittsburgh area, a cultural event recommender was build around trust relations [8]. Friends could be explicitly and implicitly rated for trust ranging from 'trust strongly' to 'block'. A recommender system for academic events [7] focused more on social network analysis (SNA) in combination with collaborative filtering (CF) and finally Cornelis et al. [3] described a hybrid event recommendation approach where both aspects of CF and content-based algorithms were employed. To our knowledge however, event recommendation algorithms were never compared in a user-centric designed experiment with a focus on optimal user satisfaction.

For a comparison of algorithms often offline metrics like RMSE, MAE or precision and recall are calculated. These kinds of metrics allow automated and objective comparison of the accuracy of the algorithms but they alone can not guarantee user satisfaction in the end [9]. As shown in [2], the use of different offline metrics can even lead to a different outcome of the 'best' algorithm for the job. Hayes et al. [6] state that real user satisfaction can only be measured in an online context. We want to improve the user satisfaction for real-life users of the event website and are therefore opting for an online user-centric evaluation of different recommendation algorithms.

2. EXPERIMENT SETUP

To find the recommendation algorithm that results in the highest user satisfaction, we have set up a user-centric evaluation experiment. For a period of 41 days, we monitored both implicit and explicit user feedback in the form of user interactions with the event website. We used the collected feedback as input for 5 different recommendation algorithms, each of which generated a list of recommendations for every user. Bollen et al. [1] hypothesizes that a set of somewhere between seven and ten items would be ideal in the sense that it can be quite varied but still manageable for the users. The users therefore received a randomly chosen recommendation list containing 8 events together with an online questionnaire. They were asked to rate different aspects about the quality of their given recommendations.

In the following subsections, we elaborate on the specifics of the experiment such as the feedback collection, the rec-

Feedback activity		Feedback value
Click on	'I like this'	1.0
Share on	Facebook/Twitter	0.9
Click on	Itinerary	0.6
Click on	Print	0.6
Click on	'Go by bus/train'	0.6
Click on	'Show more details'	0.5
Click on	'Show more dates'	0.5
Mail to	a friend	0.4
Browse to	an event	0.3

Table 1: The distinct activities that were collected as user feedback together with the feedback value indicating the interest of an individual user for a specific event ranging from 1.0 (very interested) to 0.3 (slightly interested).

ommendation algorithms, how we randomized the users, and the questionnaire.

2.1 Feedback collection

Feedback collection is a very important aspect of the recommendation process. Since the final recommendations can only be as good as the quality of their input, collecting as much high quality feedback as possible is of paramount importance. Previous feedback experiments we ran on the website [4] showed that collecting explicit feedback (in the form of explicit ratings) is very hard, since users do not rate often. Clicking and browsing through the event information pages are on the other hand activities that were abundantly logged. For optimal results, we ultimately combined implicit and explicit user feedback gathered during the run of the experiment.

Since explicit ratings are typically provided after an event has been visited, algorithms based on collaborative filtering would be useless. It therefore makes sense to utilize also implicit feedback indicators like printing the event's information, which can be collected before the event has taken place. In total 11 distinct feedback activities were combined into a feedback value that expressed the interest of a user for a specific event.

The different activities are listed in Table 1 together with their resulting feedback values which were intuitively determined. The $max()$ function is used to accumulate multiple feedback values in case a user provided feedback in more than one way for the same event.

2.2 Recommendation Algorithms

To assess the influence of the recommendation algorithm on the experience of the end-user, 5 different algorithms are used in this experiment. Each user, unaware of the different algorithms, is randomly assigned to one of the 5 groups receiving recommendations generated by one of these algorithms as described in Section 2.3.

As a baseline suggestion mechanism, the random recommender (RAND), which generates recommendations by performing a random sampling of the available events, is used. The only requirement of these random recommendations is that the event is still available (i.e. it is still possible for the user to attend the event). The evaluation of these random recommendations allows to investigate if users can distinguish random events from personalized recommendations, and if so, the relative (accuracy) improvement of more intelligent algorithms over random recommendations.

Metadata field	Weight
Artist	1.0
Category	0.7
Keyword	0.2

Table 2: The metadata fields used by the content-based recommendation algorithm with their weights indicating their relative importance.

Because of its widespread use and general applicability, standard collaborative filtering (CF) is chosen as the second algorithm of the experiment. We opted for the user-based nearest neighbor version of the algorithm (UBCF) because of the higher user-user overlap compared to the item-item overlap. Neighbors were defined as being users with a minimum overlap of 1 event in their feedback profiles but had to be at least 5% similar according to the cosine similarity metric.

The third algorithm evaluated in this experiment is singular value decomposition (SVD) [11], a well-known matrix factorization technique that addresses the problems of synonymy, polysemy, sparsity, and scalability for large datasets. Based on preceding simulations on an offline dataset with historical data of the website, the parameters of the algorithm were determined: 100 initial steps were used to train the model and the number of features was set at 70.

Considering the transiency of events and the ability of content-based (CB) algorithms to recommend items before they received any feedback, a CB algorithm was chosen as the fourth algorithm. This algorithm matches the event metadata, which contain the title, the categories, the artist(s), and keywords originating from a textual description of the event, to the personal preferences of the user, which are composed by means of these metadata and the user feedback gathered during the experiment. A weighting value is assigned to the various metadata fields (see Table 2), thereby attaching a relative importance to the fields during the matching process (e.g., a user preference for an artist is more important than a user preference for a keyword of the description). The employed keyword extraction mechanism is based on a term frequency-inverse document frequency (tf-idf) weighting scheme, and includes features as stemming and filtering stop words.

Since pure CB algorithms might produce recommendations with a limited diversity [9], and CF techniques might produce suboptimal results due to a large amount of unrated items (cold start problem), a hybrid algorithm (CB+UBCF), combining features of both CB and CF techniques, completes the list. This fifth algorithm combines the best personal sugestions produced by the CF with the best sugestions originating from the CB algorithm, thereby generating a merged list of hybrid recommendations for every user. This algorithm acts on the resulting recommendation lists produced by the CF and CB recommender, and does not change the internal working of these individual algorithms. Both lists are interwoven while alternately switching their order such that both lists have their best recommendation on top in 50% of the cases.

For each algorithm, the final event recommendations are checked for their availability and familiarity with the user.

Events that are not available for attendance anymore, or events that the user has already explored (by viewing the webpage, or clicking the link) are replaced in the recommendation list.

2.3 Randomizing Users

Since certain users have provided only a limited amount of feedback during the experiment, not all recommendation algorithms were able to generate personal suggestions for these users. CF algorithms, for instance, can only identify neighbors for users who have overlapping feedback with other users (i.e. provided feedback on the same event as another user). Without these neighbors, CF algorithms are not able to produce recommendations. Therefore, users with a limited profile, hindering (some of) the algorithms to generate (enough) recommendations for that user, are treated separately in the analysis. Many of these users are not very active on the website or did not finish the evaluation procedure as described in Section 2.4. This group of cold-start users received recommendations from a randomly assigned algorithm that was able to generate recommendations for that user based on the limited profile. Since the random recommender can produce suggestions even without user feedback, at least 1 algorithm was able to generate a recommendation list for every user. The comparative evaluation of the 5 algorithms however, is based on the remaining users. Each of these users is randomly assigned to 1 of the 5 algorithms which generates personal suggestions for that user. This way, the 5 algorithms, as described in Section 2.2, are evaluated by a number of randomly selected users.

2.4 Evaluation Procedure

While prediction accuracy of ratings used to be the only evaluation criteria for recommender systems, during recent years optimizing the user experience has increasingly gained interest in the evaluation procedure. Existing research has proposed a set of criteria detailing the characteristics that constitute a satisfying and effective recommender system from the user's point of view. To combine these criteria into a more comprehensive model which can be used to evaluate the perceived qualities of recommender systems, Pu et al. have developed an evaluation framework for recommender systems [10]. This framework aims to assess the perceived qualities of recommenders such as their usefulness, usability, interface and interaction qualities, user satisfaction of the systems and the influence of these qualities on users' behavioral intentions including their intention to tell their friends about the system, the purchase of the products recommended to them, and the return to the system in the future. Therefore, we adopted (part of) this framework to measure users' subjective attitudes based on their experience towards the event recommender and the various algorithms tested during our experiment. Via an online questionnaire, test users were asked to answer 14 questions on 5-point Likert scale from "strongly disagree" (1) to "strongly agree" (5) regarding aspects as recommendation accuracy, novelty, diversity, satisfaction and trust of the system. We selected the following 8 most relevant questions for this research regarding various aspects of the event recommendation system.

Q1 The items recommended to me matched my interests.

Q2 Some of the recommended items are familiar to me.

Algorithm	*#Users*
CB	43
CB+UBCF	36
RAND	45
SVD	36
UBCF	33

Table 3: The 5 algorithms compared in this experiment and the number of users that actually completed the questionnaire about their recommendation lists.

Q4 The recommender system helps me discover new products.

Q5 The items recommended to me are similar to each other (reverse scale).

Q7 I didn't understand why the items were recommended to me (reverse scale).

Q8 Overall, I am satisfied with the recommender.

Q10 The recommender can be trusted.

Q13 I would attend some of the events recommended, given the opportunity.

3. RESULTS

We allowed all users of the event website to participate in our experiment and encouraged them to do so by means of e-mail and a banner on the site. In total 612 users responded positively to our request. After a period of feedback logging, as described in section 2.1, they were randomly distributed across the 5 recommendation algorithms which calculated for each of them a list of 8 recommendations. After the recommendations were made available on the website, users were asked by mail to fill out the accompanying online questionnaire as described in section 2.4.

Of the 612 users who were interested in the experiment, 232 actually completed the online questionnaire regarding their recommendations. After removal of fake samples (i.e., users who answered every question with the same value) and users with incomplete (feedback) profiles, 193 users remained. They had by average 22 consumptions (i.e., expressed feedback values for events) and 84% of them had 5 or more consumptions. The final distribution of the users across the algorithms is displayed in Table 3.

Figure 1 shows the averaged results of the answers provided by the 193 users in this experiment for the 8 questions we described in section 2.4 and for each algorithm.

Evaluating the answers to the questionnaire showed that the hybrid recommender (CB+UBCF) achieved the best averaged results to all questions, except for question Q5, which asked the user to evaluate the similarity of the recommendations (i.e. diversity). For question Q5 the random recommender obtained the best results in terms of diversity, since random suggestions are rarely similar to each other. The CF algorithm was the runner-up in the evaluation and achieved a second place after the hybrid recommender for almost all questions (again except for Q5, where CF was the fourth after the random recommender, the hybrid recommender and SVD).

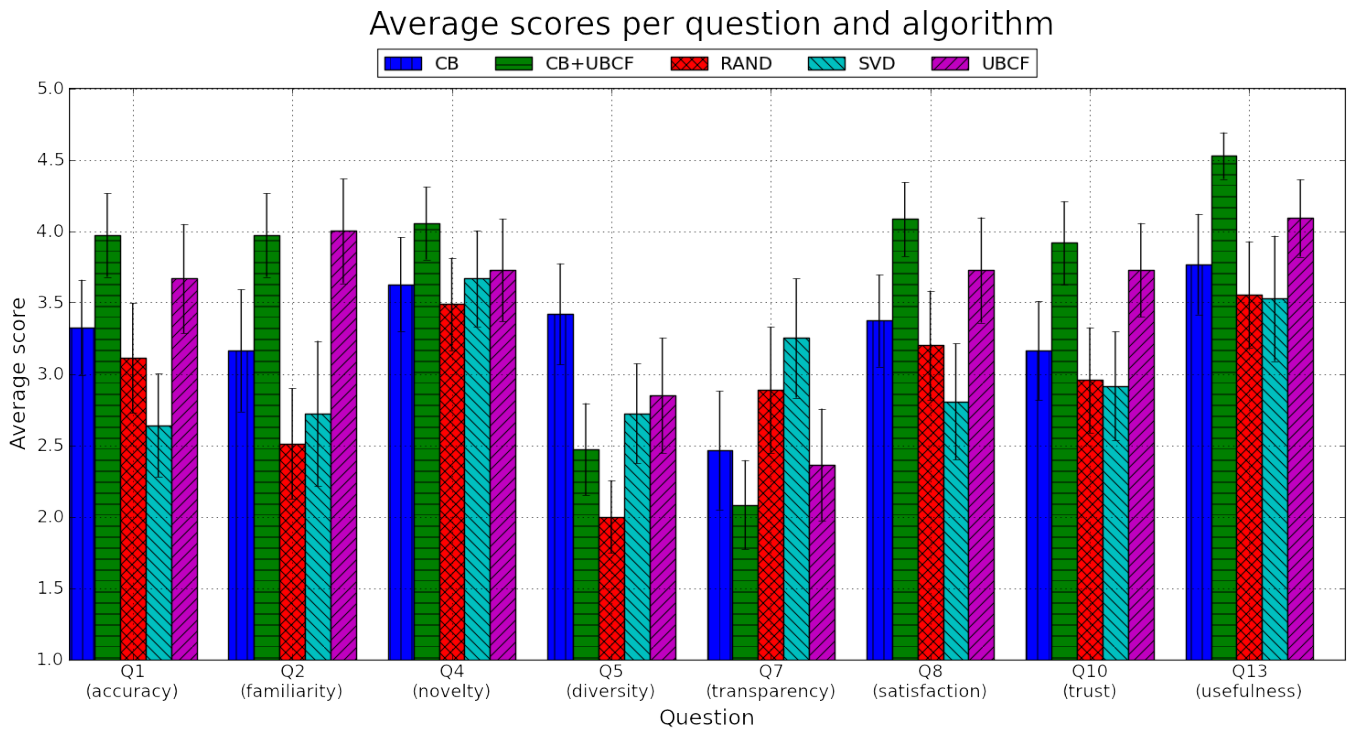

Figure 1: The averaged result of the answers (5-point Likert scale from "strongly disagree" (1) to "strongly agree" (5)) of the evaluation questionnaire for each algorithm and questions Q1, Q2, Q4, Q5, Q7, Q8, Q10 and Q13. The error bars indicate the 95% confidence interval. Note that questions Q5 and Q7 were in reverse scale.

The success of the hybrid recommenders is not only clearly visible when comparing the average scores for each question (Figure 1), but also showed to be statistically significantly better than every other algorithm (except for the CF recommender) according to a Wilcoxon rank test ($p < 0.05$) for the majority of the questions (Q1, Q2, Q8, Q10 and Q13). Table 4 shows the algorithms and questions for which statistically significant differences could be noted according to this non-parametric statistical hypothesis test.

The average performance of SVD was a bit disappointing by achieving the worst results for questions Q1, Q7, Q8, and the second worst results (after the random recommender) for questions Q2, Q4, Q10, Q11, and Q13. So surprisingly the SVD algorithm performs (averagely) worse than the random method on some fundamental questions like for example Q8 which addresses the general user satisfaction. We note however that the difference in values between SVD and the RAND algorithm was not found to be statistically significant except for question Q5.

We looked more closer into this observation and plotted a histogram (Figure 2) of the different values (1 to 5) for the answers provided for question Q8. A clear distinction between the histogram of the SVD algorithm and the histograms of the other algorithms (CB and RAND shown in the figure) can be seen. Whereas for CB and RAND most values are grouped towards one side of the histogram (i.e. the higher values), this is not the case for the SVD. It turns out that the opinions about the general satisfaction of the SVD algorithm where somewhat divided between good and bad with no apparent winning answer. These noteworthy

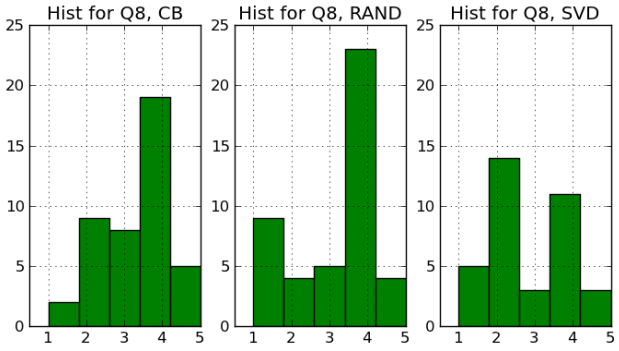

Figure 2: The histogram of the values (1 to 5) that were given to question Q8 for algorithm CB (left), RAND (middle) and SVD (right).

rating values for the SVD recommender are not only visible in the results of Q8, but also for other questions like Q2 and Q5. These findings indicate that SVD works well for many users, but also provides inaccurate recommendations for a considerable number of other users. These inaccurate recommendations may be due to a limited amount of user feedback and therefore sketchy user profiles.

Figure 1 seems to indicate that some of the answers to the questions are highly correlated. One clear example is question Q1 about whether or not the recommended items

	CB	CB+UBCF	RAND	SVD	UBCF
CB	-	Q1, Q2, Q5, Q8, Q10, Q13	Q2, Q5	Q1, Q5, Q7, Q8	Q2, Q5, Q10
CB+UBCF	Q1, Q2, Q5, Q8, Q10, Q13	-	Q1, Q2, Q4, Q5, Q7, Q8, Q10, Q13	Q1, Q2, Q7, Q8, Q10, Q13	Q13
RAND	Q2, Q5	Q1, Q2, Q4, Q5, Q7, Q8, Q10, Q13	-	Q5	Q2, Q5, Q10
SVD	Q1, Q5, Q7, Q8	Q1, Q2, Q7, Q8, Q10, Q13	Q5	-	Q1, Q2, Q7, Q8, Q10
UBCF	Q2, Q5, Q10	Q13	Q2, Q5, Q10	Q1, Q2, Q7, Q8, Q10	-

Table 4: The complete matrix of statistically significant differences between the algorithms on all the questions using the Wilcoxon rank test on a confidence level of 0.95. Note that the matrix is symmetric.

matched the user's interest and question Q8 which asked about the general user satisfaction. As obvious as this correlation may be, other correlated questions may not be so easy to detect by inspecting a graph with averaged results and so we calculated the complete correlation matrix for every question over all the algorithms using the two-tailed Pearson correlation metric (Table 5).

From the correlation values two similar trends can be noticed for questions Q8 and Q10 dealing with respectively the user satisfaction and trust of the system. The answers to these questions are highly correlated (very significant $p < 0.01$) with almost every other question except for Q5 (diversity). We must be careful not to confuse correlation with causality, but still data indicates the strong relation between user satisfaction and recommendation accuracy and transparency.

This strong relation may be another reason why SVD performed very badly in the experiment. Its inner workings are the most obscure and least obvious to the user and therefore also the least transparent.

Another interesting observation lies in the correlation values of question Q5. The answers to this diversity question are almost completely unrelated to every other question (i.e., low correlation values which are not significant $p > 0.05$). It seems like the users of the experiment did not value the diversity of a recommendation list as much as the other aspects of the recommendation system. If we look at the average results (Figure 1) of the diversity question (lower is more diverse) we can see this idea confirmed. The ordering of how diverse the recommendation lists produced by the algorithms were, is in no way reflected in the general user satisfaction or trust of the system.

To gain some deeper insight into the influence of the qualitative attributes towards each other, we performed a simple linear regression analysis. By trying to predict an attribute by using all the other ones as input to the regression function, a hint of causality may be revealed. As regression method we used multiple stepwise regression. We used a combination of the forward and backward selection approach, which step by step tries to add new variables (or remove existing ones) to its model that have the highest marginal relative influence on the dependent variable. The following lines express the regression results. We indicated what attributes were added to the model by means of an arrow notation. Between brackets we also indicated the coefficient of determination R^2. This coefficient indicates what percentage of the variance in the dependent variable can be explained by the model. R^2 will be 1 for a perfect fit and 0 when no linear relationship could be found.

Q1 $\leftarrow$ Q7, Q8, Q10, Q13 ($R^2 = 0.7131$)

Q2 $\leftarrow$ Q7, Q10, Q13 ($R^2 = 0.2195$)

Q4 $\leftarrow$ Q10, Q13 ($R^2 = 0.326$)

Q5 $\leftarrow$ Q1, Q13 ($R^2 = 0.02295$)

Q7 $\leftarrow$ Q1, Q2, Q8, Q10 ($R^2 = 0.6095$)

Q8 $\leftarrow$ Q1, Q7, Q10, Q13 ($R^2 = 0.747$)

Q10 $\leftarrow$ Q1, Q2, Q4, Q7, Q8, Q13 ($R^2 = 0.7625$)

Q13 $\leftarrow$ Q1, Q2, Q4, Q5, Q8, Q10 ($R^2 = 0.6395$)

The most interesting regression result is the line were Q8 (satisfaction) is predicted by Q1, Q7, Q10 and Q13. This result further strengthens our belief that accuracy (Q1) and transparency (Q7) are the main influencers of user satisfaction in our experiment (we consider Q10 and Q13 rather as results of satisfaction than real influencers but they are of course also connected).

Table 6 shows the coverage of the algorithms in terms of the number of users it was able to produce recommendations for. In our experiment we noticed an average coverage of 66% excluding the random recommender.

Algorithm	Coverage (%)
CB	69%
CB+UBCF	66%
RAND	100%
SVD	66%
UBCF	65%

Table 6: The 5 algorithms compared in this experiment and their coverage in terms of the number of users for which they were able to generate a recommendation list of minimum 8 items.

Next to this online and user-centric experiment, we also ran some offline tests and compared them to the real opinions of the users. We calculated the recommendations on a training set that randomly contained 80% of the collected feedback in the experiment. Using the leftover 20% as the

	Q1 (accuracy)	Q2 (familiarity)	Q4 (novelty)	Q5 (diversity)	Q7 (tranparency)	Q8 (satisfaction)	Q10 (trust)	Q13 (usefulness)
Q1	1	.431	.459	.012	-.731	.767	.783	.718
Q2	.431	1	.227	.036	-.405	.387	.429	.415
Q4	.459	.227	1	-.037	-.424	.496	.516	.542
Q5	.012	.036	-.037	1	0.16	-.008	.001	-.096
Q7	-.731	-.405	-.424	.016	1	-.722	-.707	-.622
Q8	.767	.387	.496	-.008	-.722	1	.829	.712
Q10	.783	.429	.516	.001	-.707	.829	1	.725
Q13	.718	.415	.542	-.096	-.622	.712	.725	1

Table 5: The complete correlation matrix for the answers to the 8 most relevant questions on the online questionnaire. The applied metric is the Pearson correlation and so values are distributed between -1.0 (negatively correlated) and 1.0 (positively correlated). Note that the matrix is symmetric and questions Q5 and Q7 were in reverse scale.

Algorithm	Precision (%)	Recall (%)	F1 (%)
CB	0.462	2.109	0.758
CB+UBCF	1.173	4.377	1.850
RAND	0.003	0.015	0.005
SVD	0.573	2.272	0.915
UBCF	1.359	4.817	2.119

Table 7: The accuracy of the recommendation algorithms in terms of precision, recall and F1-measure based on an offline analysis.

test set, the accuracy of every algorithm was calculated over all users in terms of precision, recall and F1-measure (Table 7). This procedure was repeated 10 times to average out any random effects.

By comparing the offline and online results in our experiment we noticed a small change in the ranking of the algorithms. In terms of precision the UBCF approach came out best followed by respectively CB+UBCF, SVD, CB and RAND. While the hybrid approach performed best in the online analysis, this is not the case for the offline tests. Note that also SVD and CB have swapped places in the ranking. SVD showed slightly better at predicting user behaviour than the CB algorithm. A possible explanation (for the inverse online results) is that users in the online test may have valued the transparency of the CB algorithm over its (objective) accuracy. Our offline evaluation test further underlines the shortcomings of these procedures. In our experiment we had over 30,000 items that were available for recommendation and on average only 22 consumptions per user. The extreme low precision and recall values are the result of this extreme sparsity problem.

It would have been interesting to be able to correlate the accuracy values obtained by offline analysis with the subjective accuracy values provided by the users. Experiments however showed very fluctuating results with on the one hand users with close to zero precision and on the other hand some users with relative high precision values. These results could therefore not be properly matched against the online gathered results.

4. DISCUSSION

The results clearly indicate the hybrid recommendation algorithm (CB+UBCF) as the overall best algorithm for optimizing the user satisfaction in our event recommendation system. The runner-up for this position would definitely be the UBCF algorithm followed by the CB algorithm. This comes as no surprise considering that the hybrid algorithm is mere a combination of these UBCF and CB algorithms. Since the UBCF algorithm is second best, it looks like this algorithm is the most responsible for the success of the hybrid. While the weights of both algorithms were equal in this experiment (i.e., the 4 best recommendations of each list were selected to be combined in the hybrid list), it would be interesting to see how the results evolve if these weights would be tuned more in favour of the CF approach (e.g., $5 * UBCF + 3 * CB$).

Because we collected both implicit and explicit feedback to serve as input for the recommendation algorithms, there were no restrictions as to what algorithms we were able to use. Implicit feedback that was logged before an event took place allowed the use of CF algorithms and the availability of item metadata enabled content-based approaches. Only in this ideal situation a hybrid CB+UBCF algorithm can serve an event recommendation system.

The slightly changed coverage is another issue that may come up when a hybrid algorithm like this is deployed. While the separate CB and UBCF algorithms had respectively coverages of 69% and 65%, the hybrid combination served 66% of the users. We can explain this increase of 1% towards the UBCF by noting that the hybrid algorithm requires a minimum of only 4 recommendations (versus 8 normally) to be able to provide the users with a recommendation list.

5. CONCLUSIONS

For a Belgian cultural events website we wanted to find a recommendation algorithm that improves the user experience in terms of user satisfaction and trust. Since offline evaluation metrics are inadequate for this task, we have set up an online and user-centric evaluation experiment with 5 popular and common recommendation algorithms i.e. CB, CB+UBCF, RAND, SVD and UBCF. We logged both implicit and explicit feedback data in the form of weighted user interactions with the event website over a period of 41 days. We extracted the users for which every algorithm was able to generate at least 8 recommendations and presented each of these users with a recommendation list randomly chosen from one of the 5 recommendation algorithms. Users were asked to fill out an online questionnaire that addressed qualitative aspects of their recommendation lists including accuracy, novelty, diversity, satisfaction, and trust.

Results clearly showed that the CB+UBCF algorithm, which is a combination of both the recommendations of CB and UBCF, outperforms (or is equally as good in the case of question Q2 and the UBCF algorithm) every other algorithm except for the diversity aspect. In terms of diversity the random recommendations turned out best, which of course makes perfectly good sense. Inspection of the correlation values between the answers of the questions revealed however that diversity is in no way correlated with user satisfaction, trust or for that matter any other qualitative aspect we investigated. The recommendation accuracy and transparency on the other hand were the two qualitative aspects highest correlated with the user satisfaction and showed promising predictors in the regression analysis.

The SVD algorithm came out last in the ranking of the algorithms and was statistically even indistinguishable from the random recommender for most of the questions except for again the diversity question (Q5). A histogram of the values for SVD and question Q8 puts this into context by revealing an almost black and white opinion pattern expressed by the users in the experiment.

6. FUTURE WORK

While we were able to investigate numerous different qualitative aspect about each algorithm individually, the experiment did not allow us, apart from indicating a best and worst algorithm, to construct an overall ranking of the recommendation algorithms. Each user ended up evaluating just one algorithm. As our future work, we intend to extend this experiment with a focus group allowing to elaborate on the reasoning behind some of the answers users provided and compare subjective rankings of the algorithms.

We also plan to extend our regression analysis to come up with a causal path model that will allow us to have a better understanding as to how the different algorithms influence the overall satisfaction.

7. ACKNOWLEDGMENTS

The research activities that have been described in this paper were funded by a PhD grant to Simon Dooms of the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT Vlaanderen) and a PhD grant to Toon De Pessemier of the Fund for Scientific Research-Flanders (FWO Vlaanderen). We would like to thank CultuurNet Vlaanderen for the effort and support they were willing to provide for deploying the experiment described in this paper.

8. REFERENCES

[1] D. Bollen, B. Knijnenburg, M. Willemsen, and M. Graus. Understanding choice overload in recommender systems. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 63–70. ACM, 2010.

[2] E. Campochiaro, R. Casatta, P. Cremonesi, and R. Turrin. Do metrics make recommender algorithms? In *Proceedings of the 2009 International Conference on Advanced Information Networking and Applications Workshops*, WAINA '09, pages 648–653, Washington, DC, USA, 2009. IEEE Computer Society.

[3] C. Cornelis, X. Guo, J. Lu, and G. Zhang. A fuzzy relational approach to event recommendation. In *Proceedings of the Indian International Conference on Artificial Intelligence*, 2005.

[4] S. Dooms, T. De Pessemier, and L. Martens. An online evaluation of explicit feedback mechanisms for recommender systems. In *Proceedings of the 7th International Conference on Web Information Systems and Technologies (WEBIST)*, 2011.

[5] X. Guo, G. Zhang, E. Chew, and S. Burdon. A hybrid recommendation approach for one-and-only items. *AI 2005: Advances in Artificial Intelligence*, pages 457–466, 2005.

[6] C. Hayes, P. Massa, P. Avesani, and P. Cunningham. An on-line evaluation framework for recommender systems. In *Workshop on Personalization and Recommendation in E-Commerce*. Citeseer, 2002.

[7] R. Klamma, P. Cuong, and Y. Cao. You never walk alone: Recommending academic events based on social network analysis. *Complex Sciences*, pages 657–670, 2009.

[8] D. Lee. Pittcult: trust-based cultural event recommender. In *Proceedings of the 2008 ACM conference on Recommender systems*, pages 311–314. ACM, 2008.

[9] S. McNee, J. Riedl, and J. Konstan. Being accurate is not enough: how accuracy metrics have hurt recommender systems. In *CHI'06 extended abstracts on Human factors in computing systems*, page 1101. ACM, 2006.

[10] P. Pu and L. Chen. A user-centric evaluation framework of recommender systems. In *Proc. ACM RecSys 2010 Workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces (UCERSTI)*, 2010.

[11] B. Sarwar, G. Karypis, J. Konstan, J. Riedl, and M. U. M. D. O. C. SCIENCE. *Application of dimensionality reduction in recommender system-a case study*. Citeseer, 2000.